

CLAIMS

What is claimed is:

1 1. An assembler for processing structured assembly language expressions utilized in
2 structured assembly language programming, said assembler comprising:

3 program code means for recognizing a structured assembly language
4 expression's mnemonics containing elements arg1 cc arg2, wherein said cc is a
5 condition code, wherein the form of said expression's mnemonics or the nature of
6 one or more of said expression's elements selects a corresponding comparison
7 opcode, wherein said arg1 and said arg2 are valid arguments for said selected
8 comparison opcode;

9 program code means for constructing a data structure referencing said arg1,
10 said arg2, said cc, and a branch destination;

11 program code means for generating a comparison opcode in response to
12 elements of said data structure;

13 program code means for generating a conditional branch based on said
14 condition code in said data structure;

15 program code means for generating a first branch location for execution to
16 proceed at if said structured assembly language expression is true; and

17 program code means for generating a second branch location for execution
18 to proceed at if said structured assembly language expression is false; and

19 program code means for generating a third branch location for execution to
20 proceed at to the end of said structured assembly language expression; and

21 program code means for indicating said branch destination in said data
22 structure is a branch to said first, said second, or said third branch locations.

1 2. The assembler of Claim 1, wherein said assembler further includes program code
2 means for recognizing a structured assembly language expression's mnemonics having a
3 form cc, wherein said cc is a condition code.

1 3. The assembler of Claim 1, wherein said assembler further includes a program code
2 means for generating a data structure referencing at least no arguments, cc, and a branch
3 destination in response to said condition code.

1 4. The assembler of Claim 1, wherein said assembler further includes program code
2 means for not generating a comparison opcode in response to said data structure.

1 5. The assembler of Claim 1, wherein said assembler further includes a program code
2 means for generating assembly language code by iterating over a vector of said structured
3 assembly language data structures of various forms.

1 6. The assembler of Claim 1, wherein said assembler further includes

2 program code means for recognizing a structured assembly language
3 expression's mnemonics resulting from a logical ANDing of SA_Expr1 and
4 SA_Expr2, wherein each of said SA_Expr1 and said SA_Expr2 is a unit or a
5 compound structured assembly language expression;

6 program code means for setting said branch in each data structure of said
7 SA_Expr1 that is branching to said first branch location to branch to end of said
8 SA_Expr1; and

9 program code means for concatenating and preserving order of data
10 structures in said SA_Expr1 and said SA_Expr2 into a single compound structured
11 assembly language expression.

1 7. The assembler of Claim 1, wherein said assembler further includes

2 program code means for recognizing a structured assembly language
3 expression's mnemonics requiring a logical ORing of SA_Expr3 and SA_Expr4,
4 wherein each of said SA_Expr3 and said SA_Expr4 is a unit or a compound
5 structured assembly language expression;

6 program code means for changing said branch location in each of said
7 SA_Expr3's data structures, except for said SA_Expr3's last data structure, from
8 said second branch location to end of said SA_Expr3;

9 program code means for complementing said branch condition in said
10 SA_Expr3's last data structure;

11 program code means for changing said branch location in said SA_Expr3's
12 last data structure from a branch to said first location to branch to said second
13 location, or from a branch to said second location to branch to said first location;
14 and

15 program code means for concatenating and preserving order of data
16 structures in said SA_Expr3 and said SA_Expr4 into a single compound structured
17 assembly language expression.

1 8. The assembler of Claim 1, wherein said assembler further includes

2 program code means for recognizing said structured assembly language
3 expression's mnemonics requiring from a logical negation of SA_Expr5, wherein
4 said SA_Expr5 is a unit or compound structured assembly language expression;

5 program code means for changing said branch location in each of said
6 SA_Expr5's data structures, except for said SA_Expr5's last data structure from said
7 first branch location to said second branch location, while changing said branch
8 location in each of said SA_Expr5's data structures, except for said SA_Expr5's last
9 data structure, from said second branch location to said first branch location; and

10 program code means for complementing said branch condition in said
11 SA_Expr5's last data structure.